# Serial I/O and USART 8251A

The serial I/O technique is commonly used to interface terminals, printers, and modems. These peripherals and computers are designed and manufactured by various manufacturers. Therefore, a common understanding must exist, among various manufacturing and user groups, that can ensure compatibility among different equipment. When this understanding is defined and generally accepted in industry (and by users), it is known as a standard. A standard is normally defined by a professional organization (such as IEEE—Institute of Electrical and Electronics Engineers); however, occasionally, a widespread practice can become a de facto standard. A standard may include such items as assignment of pin positions for signals, voltage levels, speed of data transfer, length of cables, and mechanical specifications.

In serial I/O, data can be transmitted as either current or voltage. Typically, 20 mA (or 60 mA) current loops are used in teletype equipment. When a teletype is marking or at logic 1, current flows; when it is at logic 0 (or Space), the current flow is interrupted. The advantage of the current loop method is that signals are relatively noise-free and are suitable for transmission over a distance.

When data are transmitted as voltage, the commonly used standard is known as RS-232C. It is defined in reference to Data Terminal Equipment (DTE) and Data Communication Equipment (DCE)—terminal and modem—as shown in Figure 16.5(a); however, its voltage levels are not compatible with TTL logic levels. The rate of data transmission in RS-232C is restricted to a maximum of 20 kbaud and a distance of 50 ft. For high-speed data transmission, two new standards—RS-422A and RS-423A—have been developed in recent years; however, they are not yet widely used.

To appreciate the difficulties and confusion in this standard, one has to examine its historical background. The RS-232 Standard was developed during the initial days of computer timesharing, long before the existence of TTL logic, and its primary focus was to have compatibility between a terminal and a modem. However, the same standard is now being used for communications between computers and peripherals, and the roles of a data terminal and a modem have become ambiguous. Should a computer be considered a terminal or a modem? The answer is that it can be either. Therefore, the lines used for transmission and reception will differ, depending on the manufacturer's role-definition of its equipment.

### RS-232C

Figure 16.5(b) shows the RS-232C 25-pin connector and its signals. The signals are divided into four groups: data signals, control signals, timing signals, and grounds. For data lines, the voltage level +3 V to +15 V is defined as logic 0; from −3 V to −15 V is defined as logic 1 (normally, voltage levels are ±12 V). This is negative true logic. However, other signals (control and timing) are compatible with the TTL level. Because of incompatibility of the data lines with the TTL logic, voltage translators, called line drivers and line receivers, are required to interface TTL logic with the RS-232 signals, as shown in Figure 16.5(a). The line driver, MC1488, converts logic 1 into approximately −9 V and logic 0 into +9 V, as shown in Figure 16.5(a). Before it is received by the DCE, it is again converted by the line receiver, MC1489, into TTL-compatible logic. This raises the question: If the received signal is to be converted back to the TTL level, what is the reason, in the first place, to convert the transmitted signal to the higher level? The primary reason is

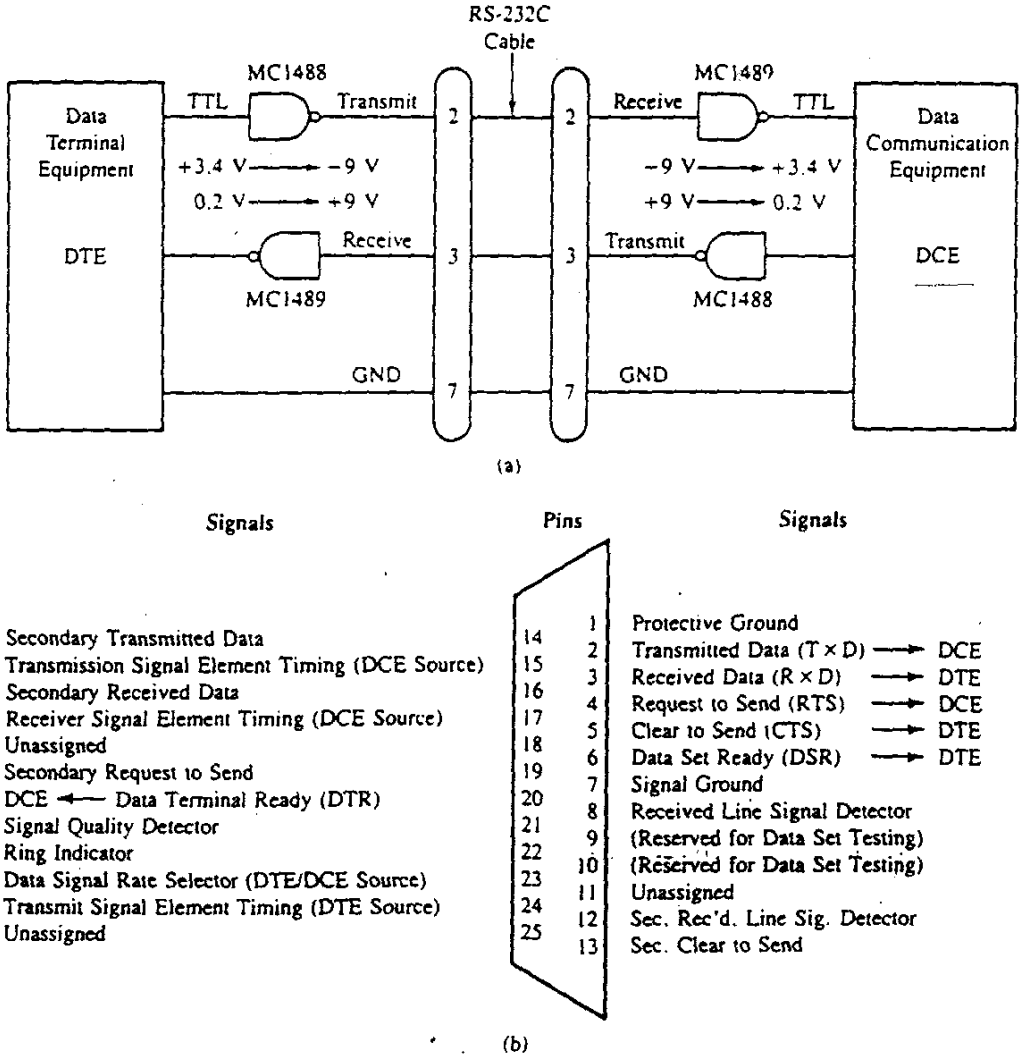



FIGURE 16.5
Minimum Configuration of RS-232C Signals and Voltage Levels (a) and RS-232C Signal Definitions and Pin Assignments (b)
SOURCE: Courtesy of Electronic Industries Association.

that the standard was defined before the TTL levels came into existence; before 1960, most equipment was designed to handle higher voltages. The other reason is that this standard provides a higher level of noise margin—from −3 V to +3 V.

The minimum interface between a computer and a peripheral requires three lines: pins 2, 3, and 7, as shown in Figure 16.5(a). These lines are defined in relation to the DTE; the terminal transmits on pin 2 and receives on pin 3. On the other hand, the DCE transmits on pin 3 and receives on pin 2. Now the dilemma is: How does a manufacturer define the role of its equipment? For example, the user may connect its microcomputer to a printer configured as a DTE or to a modem configured as a DCE. Therefore, to remain compatible with the defined signals of the RS-232C, the computer must be defined as a DCE for the printer connection and a DTE for the modem. One of the solutions to this dilemma is for the manufacturer to provide two serial ports, one for a modem and the other for a printer. Another solution is to reconfigure the RS-232 cable as shown in Figure 16.6. In Figure 16.6; the microcomputer is defined as a DTE, and it can be connected to the modem, defined as a DCE, without any modification in the RS-232 cable, as shown in Figure 16.6 (a). However, when it is connected to the printer, the transmit and the receive lines must be crossed as shown in Figure 16.6(b).

Typically, data transmission with a handshake requires eight lines, listed in Table 16.1. Specific functions of handshake lines differ in different peripherals and, therefore, should be referred to in the manufacturers' manuals.

For high-speed transmission, the standards RS-422A and RS-423A are used. These standards use differential amplifiers to reject noise levels and can transmit data at higher speed with longer cable. The RS-422A allows a maximum speed of 10 Mbaud for a 40-ft distance and 100 kbaud for 4000 ft. The RS-423A is limited to 100 kbaud for a 30-ft distance and 10 kbaud for 300 ft. See Table 16.2 for comparison of the three standards: RS-232C, RS-422A, and RS-423A.

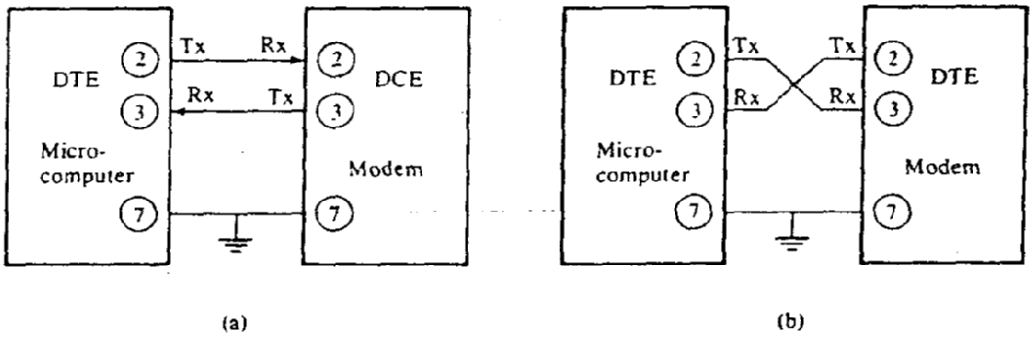



FIGURE 16.6
RS-232C Connections: DTE to DCE (a) and DTE to DTE (b)

TABLE 16.1
RS-232C Signals Used with Handshake Data Communication

| Pin No. | Signals[a] | | Functions |
|---|---|---|---|
| 2 | Transmitted Data | TxD | Output; transmits data from DTE to DCE |
| 3 | Received Data | RxD | Input; DTE receives data from DCE |
| 4 | Request to Send | RTS | General-purpose output from DTE |
| 5 | Clear to Send | CTS | General-purpose input to DTE; can be used as a handshake signal |
| 6 | Data Set Ready | DSR | General-purpose input to DTE; can be used to indicate that DCE is ready |
| 7 | Signal Ground | GND | Common reference between DTE and DCE |
| 8 | Data Carrier Detect | DCD | Generally used by DTE to disable data reception |
| 20 | Data Terminal Ready | DTR | Output; generally used to indicate that DTE is ready |

[a]Signals are referenced to DTE.

TABLE 16.2
Comparison of Serial I/O Standards

| Specifications | RS-232C | RS-422A | RS-423A |
|---|---|---|---|
| Speed | 20 kbaud | 10 Mbaud at 40 ft<br>100 kbaud at 4000 ft | 100 kbaud at 30 ft<br>1 kbaud at 4000 ft |
| Distance | 50 ft | 4000 ft | 4000 ft |
| Logic 0 | > +3 to +25 V | B > A[a] | +4 to +6 V |
| Logic 1 | < −3 to −25 V | B < A | −4 to −6 V |
| Receiver Input Voltage | ±15 V | ±7 V | ±12 V |

[a]B and A are differential input to the op amp.

## 16.41 The 8251A Progammable Communication Interface

The 8251A is a programmable chip designed for synchronous and asynchronous serial data communication, packaged in a 28-pin DIP. The 8251A is the enhanced version of its predecessor, the 8251, and is compatible with the 8251. Figure 16.12 shows the block diagram of the 8251A. It includes five sections: Read/Write Control Logic, Transmitter, Receiver, Data Bus Buffer, and Modem Control.

The control logic interfaces the chip with the MPU, determines the functions of the chip according to the control word in its register (to be explained below), and monitors the data flow. The transmitter section converts a parallel word received from the MPU into serial bits and transmits them over the TxD line to a peripheral. The receiver section receives serial bits from a peripheral, converts them into a parallel word, and transfers the word to the MPU. The modem control is used to establish data communication through modems over telephone lines. The 8251A is a complex device, capable of performing various functions. For the sake of clarity, this chapter focuses only on the asynchronous mode of serial I/O and excludes any discussion of the synchronous mode and the modem control. The asynchronous mode is often used for data communication between the MPU and serial peripherals such as terminals and floppy disks.

Figure 16.13 shows an expanded version of the 8251A block diagram. The block diagram shows all the elements of a programmable chip; it includes the interfacing signals, the control register, and the status register. The functions of various blocks are described below.

### READ/WRITE CONTROL LOGIC AND REGISTERS

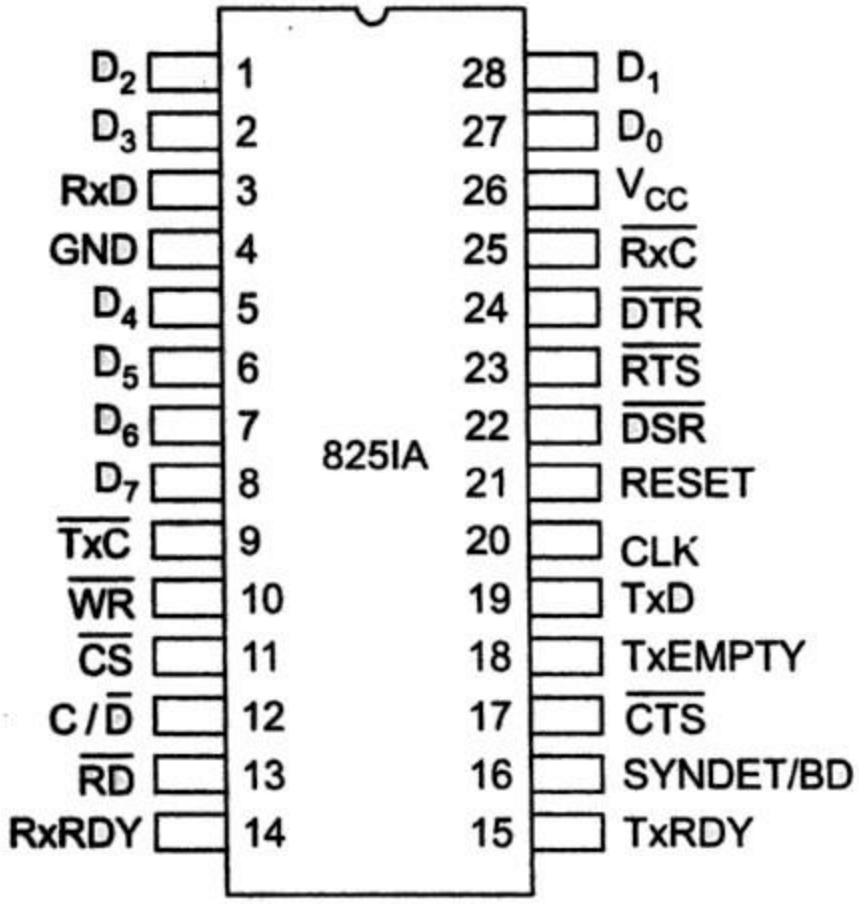
This section includes R/W control logic, six input signals, control logic, and three buffer registers: data register, control register, and status register. The input signals to the control logic are as follows.

#### Input Signals

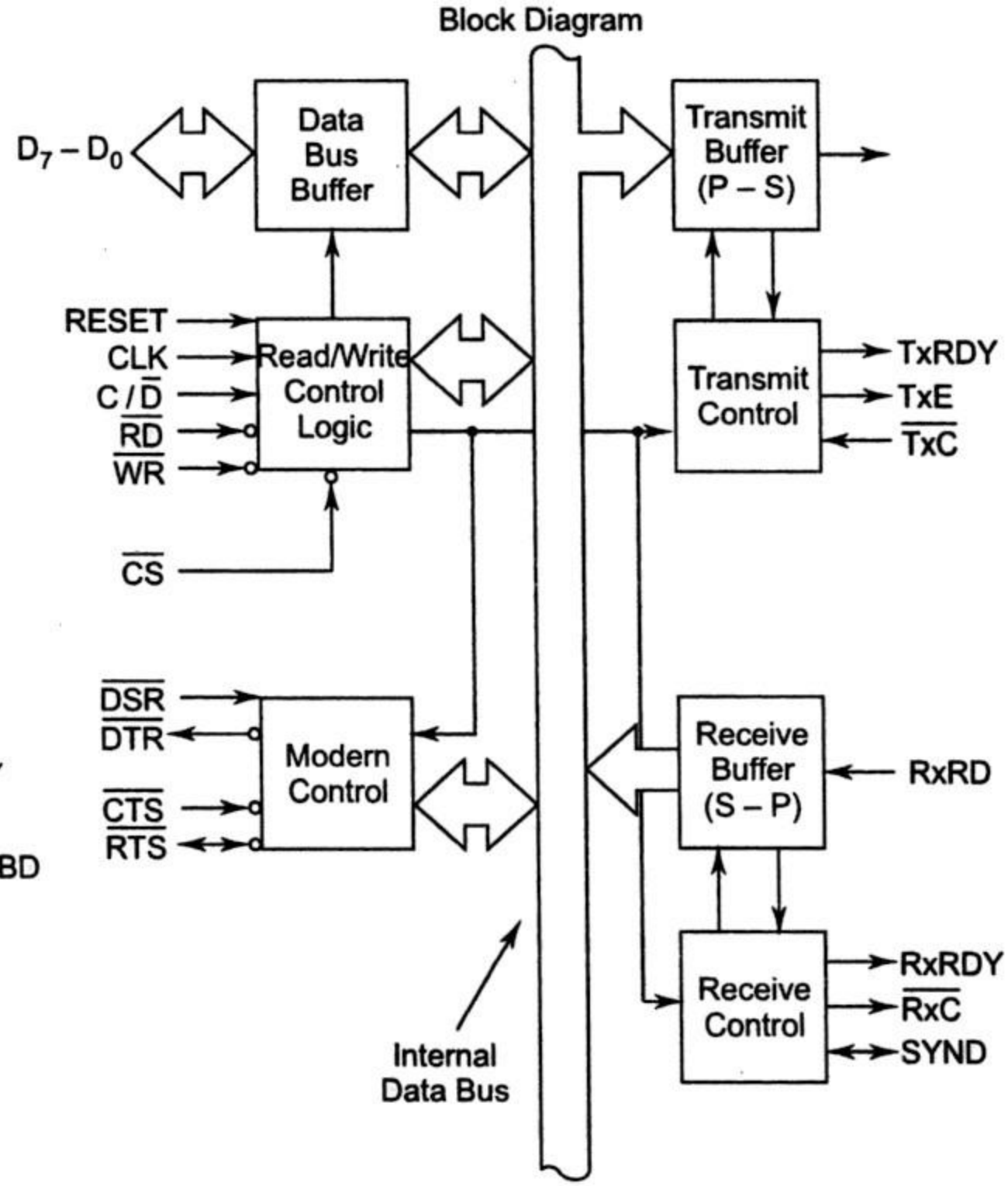
- $\overline{CS}$—**Chip Select:** When this signal goes low, the 8251A is selected by the MPU for communication. This is usually connected to a decoded address bus.
- C/$\overline{D}$—**Control/Data:** When this signal is high, the control register or the status register is addressed; when it is low, the data buffer is addressed. The control register and the status register are differentiated by $\overline{WR}$ and $\overline{RD}$ signals, respectively.

- $\overline{WR}$—**Write:** When this signal goes low, the MPU either writes in the control register or sends output to the data buffer. This is connected to $\overline{IOW}$ or $\overline{MEMW}$.
- $\overline{RD}$—**Read:** When this signal goes low, the MPU either reads a status from the status register or accepts (inputs) data from the data buffer. This is connected to either $\overline{IOR}$ or $\overline{MEMR}$.
- **RESET—Reset:** A high on this input resets the 8251A and forces it into the idle mode.
- **CLK—Clock:** This is the clock input, usually connected to the system clock. This clock does not control either the transmission or the reception rate. The clock is necessary for communication with the microprocessor.

Pin Configuration




Block Diagram




Pin Names

| | |
|---|---|
| $D_7 - D_0$ | Data Bus (8 bits) |
| C/D | Control or Data is to be Written or Read |
| RD | Read Data Command |
| WR | Write Data or Control Command |
| CS | Chip Enable |
| CLK | Clock Pulse (TTL) |
| RESET | Reset |
| TxC | Transmitter Clock |
| TxD | Transmitter Data |
| RxC | Receiver Clock |
| RxD | Receiver Data |
| RxRDY | Receiver Ready |
| TxRDY | Transmitter Ready |

| | |
|---|---|
| DSR | Data Set Ready |
| DTR | Data Terminal Ready |
| SYNDET/BD | Sync Detect/ Break Detect |
| RTS | Request to Send Data |
| CTS | Clear to Send Data |
| TxE | Transmitter Empty |
| $V_{CC}$ | + 5 Volt Supply |
| GND | Ground |

**Fig. 16.12** *The 8251A : Block Diagram, Pin Configuration, and Description*

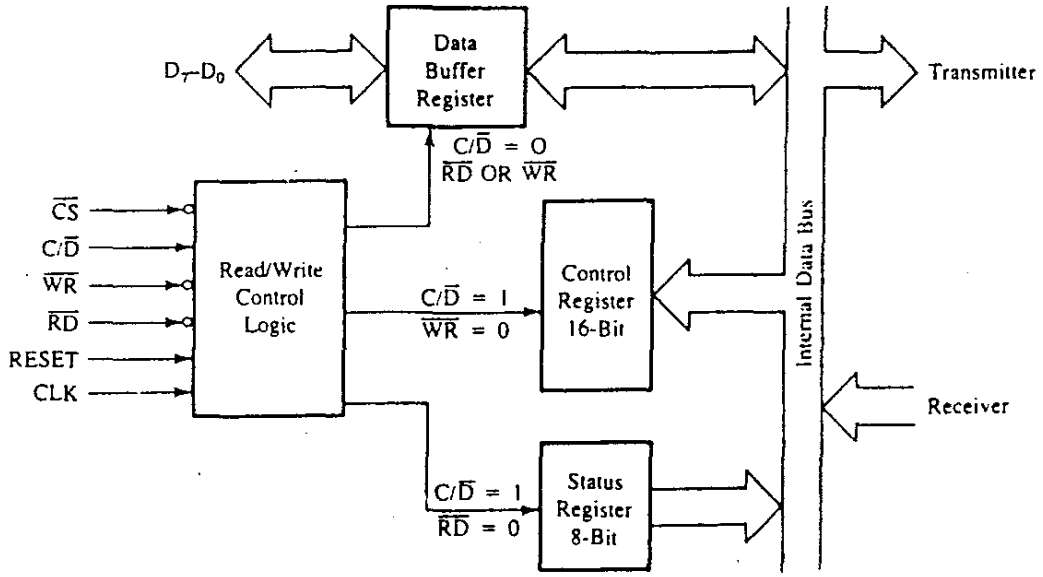



FIGURE 16.13
The 8251A: Expanded Block Diagram of Control Logic and Registers

**Control Register** This 16-bit register for a control word consists of two independent bytes: the first byte is called the **mode instruction** (word) and the second byte is called **command instruction** (word). This register can be accessed as an output port when the C/$\overline{D}$ pin is high.

**Status Register** This input register checks the ready status of a peripheral. This register is addressed as an input port when the C/$\overline{D}$ pin is high; it has the same port address as the control register.

**Data Buffer** This bidirectional register can be addressed as an input port and an output port when the C/$\overline{D}$ pin is low. Table 16.4 summarizes all the interfacing and control signals.

**TABLE 16.4**
Summary of Control Signals for the 8251A

| $\overline{CS}$ | C/$\overline{D}$ | $\overline{RD}$ | $\overline{WR}$ | Function |
|---|---|---|---|---|
| 0 | 1 | 1 | 0 | MPU writes instructions in the control register |
| 0 | 1 | 0 | 1 | MPU reads status from the status register |
| 0 | 0 | 1 | 0 | MPU outputs data to the Data Buffer |
| 0 | 0 | 0 | 1 | MPU accepts data from the Data Buffer |
| 1 | X | X | X | USART is not selected |

## TRANSMITTER SECTION

The transmitter accepts parallel data from the MPU and converts them into serial data. It has two registers: a buffer register to hold eight bits and an output register to convert eight bits into a stream of serial bits (Figure 16.14). The MPU writes a byte in the buffer register; whenever the output register is empty, the contents of the buffer register are transferred to the output register. This section transmits data on the TxD pin with the appropriate framing bits (Start and Stop). Three output signals and one input signal are associated with the transmitter section.

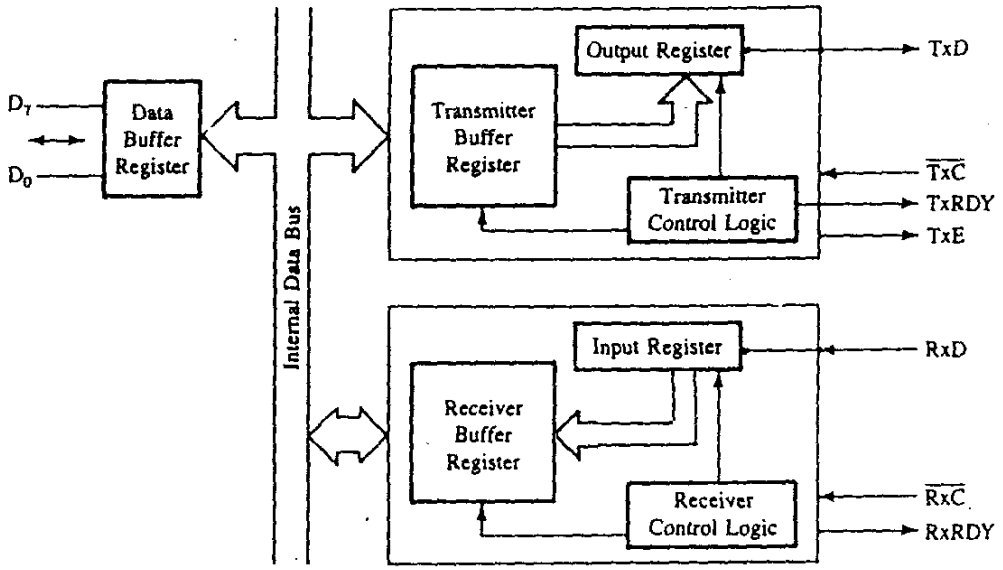



FIGURE 16.14
The 8251A: Expanded Block Diagram of Transmitter and Receiver Sections

- TxD—Transmit Data: Serial bits are transmitted on this line.
- $\overline{TxC}$—Transmitter Clock: This input signal controls the rate at which bits are transmitted by the USART. The clock frequency can be 1, 16, or 64 times the baud.
- TxRDY—Transmitter Ready: This is an output signal. When it is high, it indicates that the buffer register is empty and the USART is ready to accept a byte. It can be used either to interrupt the MPU or to indicate the status. This signal is reset when a data byte is loaded into the buffer.
- TxE—Transmitter Empty: This is an output signal. Logic 1 on this line indicates that the output register is empty. This signal is reset when a byte is transferred from the buffer to the output registers.

## RECEIVER SECTION

The receiver accepts serial data on the RxD line from a peripheral and converts them into parallel data. The section has two registers: the receiver input register and the buffer register (Figure 16.14).

When the RxD line goes low, the control logic assumes it is a Start bit, waits for half a bit time, and samples the line again. If the line is still low, the input register accepts the following bits, forms a character, and loads it into the buffer register. Subsequently, the parallel byte is transferred to the MPU when requested. In the asynchronous mode, two input signals and one output signal are necessary, as described below.

- RxD—Receive Data: Bits are received serially on this line and converted into a parallel byte in the receiver input register.
- $\overline{RxC}$—Receiver Clock: This is a clock signal that controls the rate at which bits are received by the USART. In the asynchronous mode, the clock can be set to 1, 16, or 64 times the baud.
- RxRDY—Receiver Ready: This is an output signal. It goes high when the USART has a character in the buffer register and is ready to transfer it to the MPU. This line can be used either to indicate the status or to interrupt the MPU.

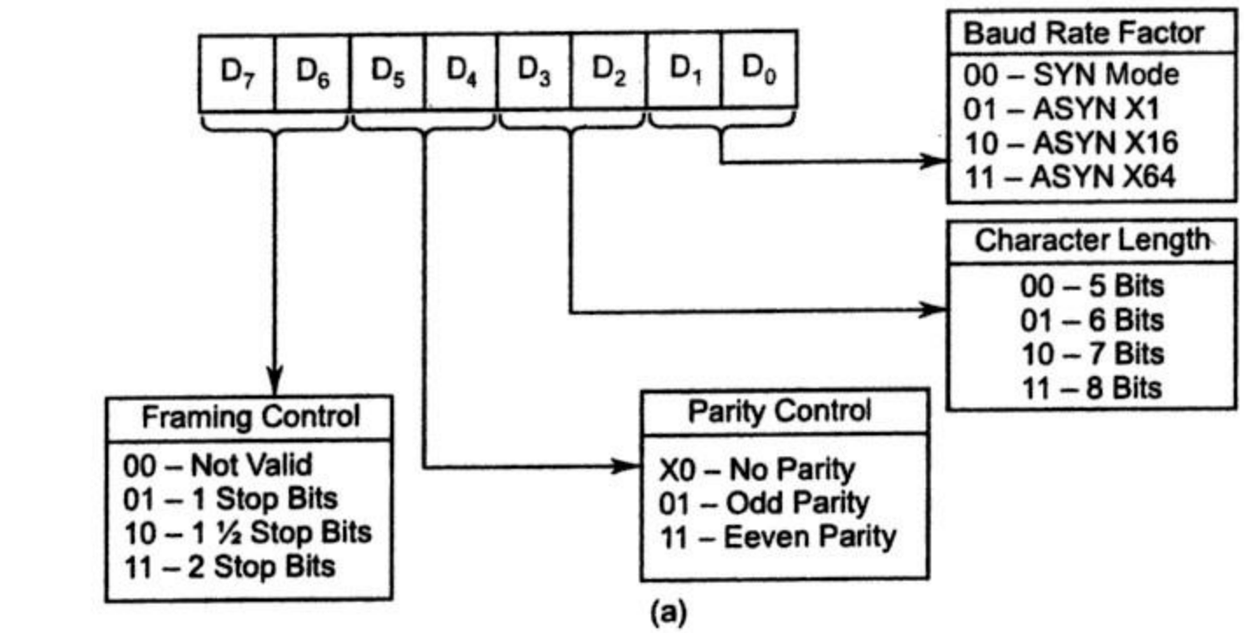
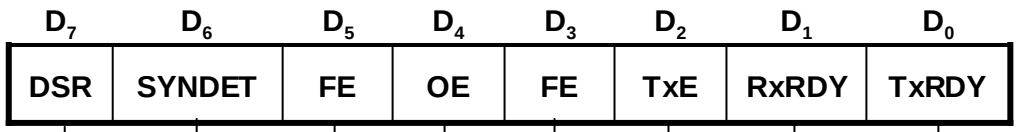



**Fig. 16.15** *Mode Word Format (a), Command Word Format (b), and Status Word Format (c)*

# 8251A Status Word




**Data Set READY**
DSR is general purpose. Normally used to test modem conditions such as Data Set Ready.

**SYNC DETECT**
When set for internal sync detect indicates that character sync has been achieved and 8251 is ready for data.

**Framing Error**
**(Asynchronous Only)**
FE flag is set when a valid stop bit is not detected at end of each character. It is reset by ER bit of Command instruction. FE does not inhibit operation of 8251.

**OVERRUN ERROR**
The OE flag is set when the CPU does not read a character before the next one becomes available. It is reset by the ER bit of the Command instruction. OE does not inhibit operation of 8251; however the previously overrun character is lost.

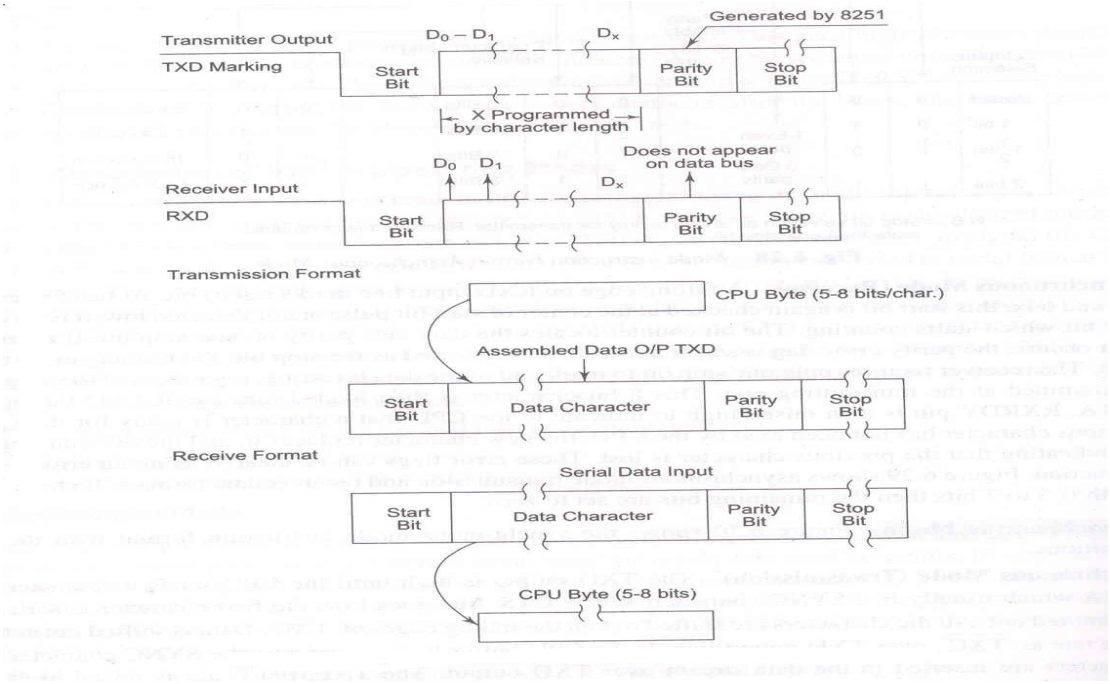
**PARITY ERROR**
PE flag is set when a parity error is detected. It is reset by the ER bit of the Command instruction. OE does not inhibit operation of 8251.

**TRANSMITTER READY**
Indicates USART is ready to accept a data character or command.

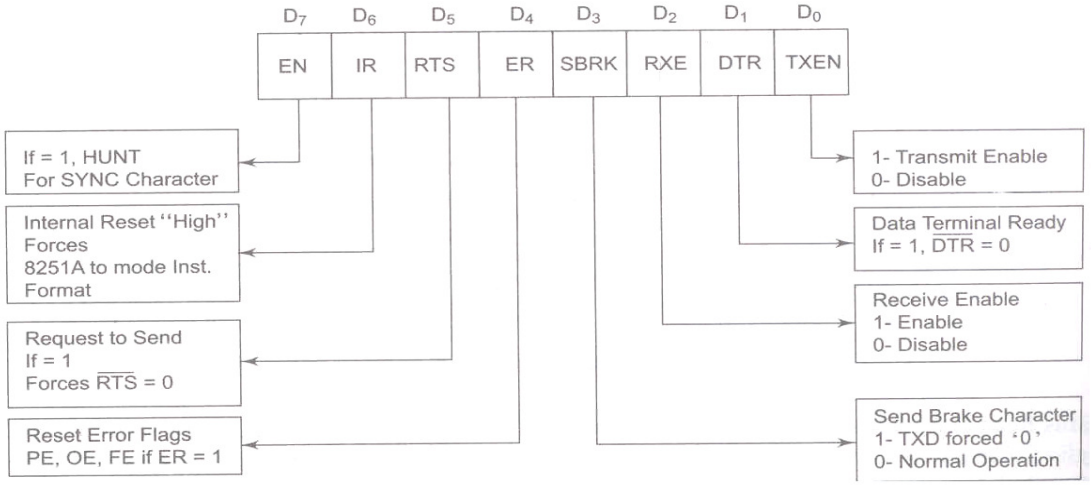
**RECEIVE READY**
Indicate USART has received a character on its serial input and is ready to transfer it to the CPU.

**TRANSMITTER EMPTY**
Indicates that parallel to serial converter in transmitter is empty.

-

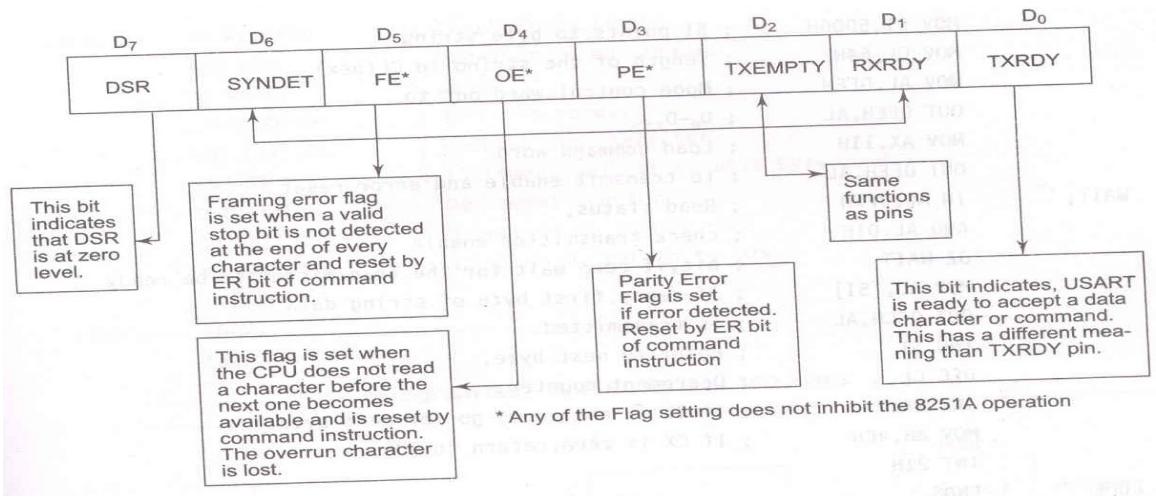



Asynchronous Mode Transmit and Receive Formats

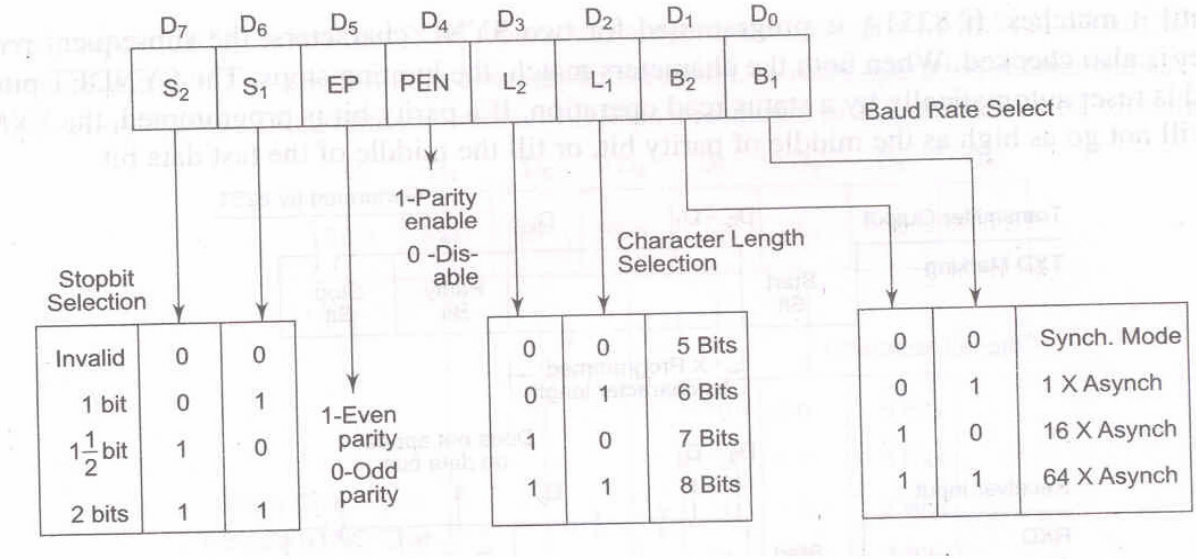



Command Instruction Format

# Interfacing 8251 with 8086




Status Read Instruction Format




Mode Instruction Format Asynchronous Mode

## INITIALIZING THE 8251A

To implement serial communication, the MPU must inform the 8251A of all details, such as mode, baud, Stop bits, parity, etc. Therefore, prior to data transfer, a set of control words must be loaded into the 16-bit control register of the 8251A. In addition, the MPU must check the readiness of a peripheral by reading the status register. The control words are divided into two formats: mode words and command words. The mode word specifies the general characteristics of operation (such as baud, parity, number of Stop bits), the command word enables data transmission and/or reception, and the status word provides the information concerning register status and transmission errors. Figure 16.15 shows the definitions of these words.

To initialize the 8251A in the asynchronous mode, a certain sequence of control words must be followed. After a Reset operation (system Reset or through instruction), a mode word must be written in the control register followed by a command word. Any control word written into the control register immediately after a mode word will be interpreted as a command word; that means a command word can be changed anytime during the operation. However, the 8251A should be reset prior to writing a new mode word, and it can be reset by using the Internal Reset bit ($D_6$) in the command word.

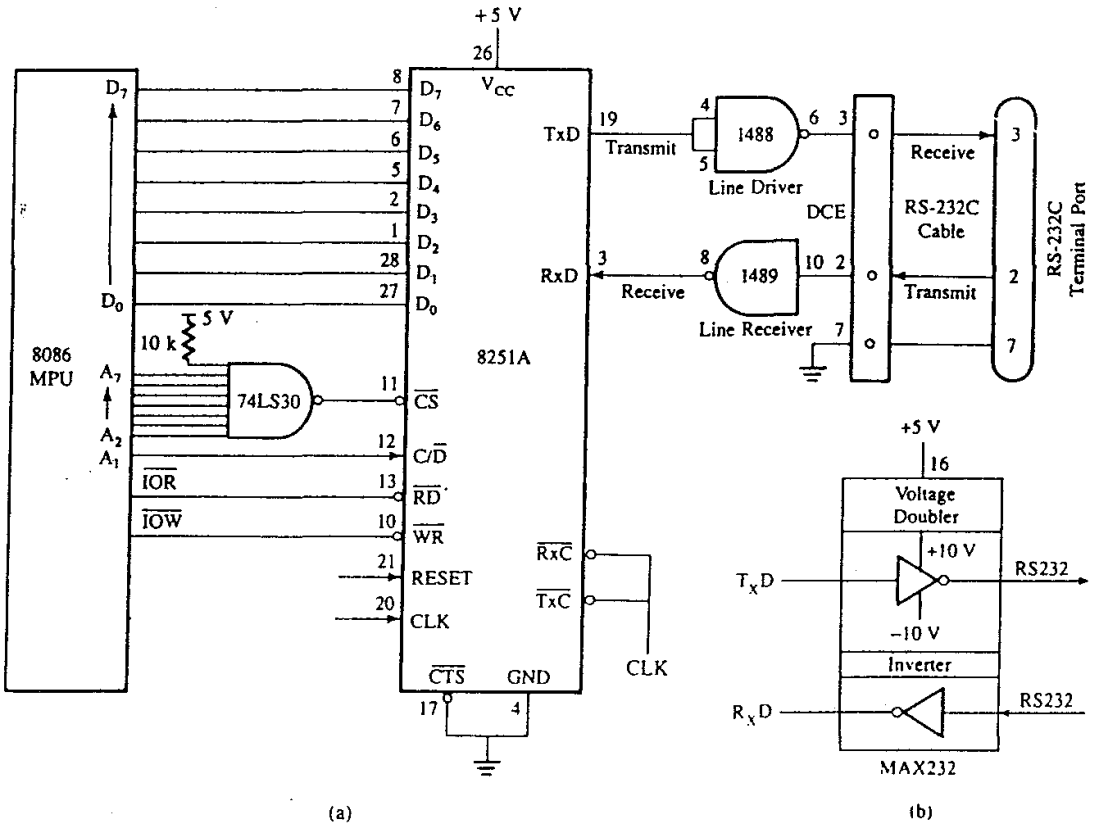



FIGURE 16.16
Schematic of Interfacing an RS-232 Terminal with an 8086 System Using the 8251A (a) and MAX 232 Logic Diagram (b)

## 16.42 Illustration: Interfacing an RS-232 Terminal Using the 8251A

PROBLEM STATEMENT

1. Identify the port addresses of the control register, the status register, and the data register in Figure 16.16.
2. Explain the RS-232 signals and the operations of the line driver (MC 1488) and the line receiver (MC 1489) shown in Figure 16.16.
3. Specify the initialization instructions and the status word to transmit characters with the following parameters if the transmitter clock frequency (TxC) is 153.6 kHz.
   - ☐ Asynchronous mode with 9600 baud
   - ☐ Character length = seven bits and two Stop bits
   - ☐ No parity check
4. Write instructions to initialize the 8251A, to read the status word, and set up a loop until the transmitter (TxRDY) is ready.

### 1. PORT ADDRESSES

a. The Chip Select line of the 8251A is enabled when the address lines $A_7$ through $A_2$ are at logic 1. To select the control register or the status register, the C/$\overline{D}$ line should be high, which means that address line $A_1$ should be 1. Therefore, the port address of the control register and the status register = FEH.

The control register is an output port and the status register is an input port; they are identified by $\overline{WR}$ and $\overline{RD}$ signals, even if their port addresses are the same.

b. The data register is selected when the C/$\overline{D}$ line goes low; thus, $A_1$ should be low. The port address of the data register = FCH. The register is bidirectional, and the same address is used to receive or transmit data. The input and output functions are identified by $\overline{RD}$ and $\overline{WR}$ signals.

### 2. RS-232C SIGNALS, LINE DRIVERS, AND LINE RECEIVERS

Figure 16.16 shows that three RS-232 signals—TxD, RxD, and Ground—are being used for serial communication between the CRT terminal and the 8086 system. The terminal transmits data on pin 2 and receives on pin 3; on the other hand, the 8086 system receives on pin 2 and transmits on pin 3 using the 8251A. Therefore, the terminal is connected as the DTE and the system plays the role of the DCE; the 8251A is part of the 8086 system.

Data transmitted over the TxD line (pin 19 of the 8251A) are at the TTL logic level. These bits are converted to RS-232 voltage levels and negative logic by line driver MC 1488. Data received by the 8251A over the RxD line (pin 3) should be at the TTL logic level. Therefore, the RS-232 signals at pin 2 of the connector are converted to the positive TTL logic level by line receiver MC 1489. The line driver and receiver are described here briefly.

**Line Driver: MC 1488** This is a quad line driver that converts TTL input levels to a maximum +15 $V_{DC}$ output signal. Typically, it is used with ±12 V power supply. For logic 0 input (< 0.8 $V_{DC}$) the output is around +10 V, and for logic 1 input (> +2.4 $V_{DC}$) the output is around −10 V; thus, the positive true logic is converted into negative true logic for RS-232C signals. The internal circuit of the MC 1488 functions much like

a comparator. For an input lower than the threshold voltage, the output approaches positive power supply voltage, and for an input higher than the threshold voltage, the output approaches negative power supply voltage.

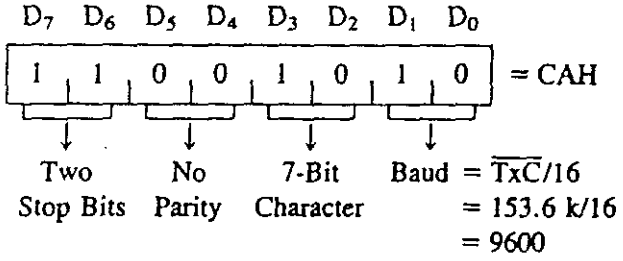
**Line Receiver: MC 1489** This is a quad line receiver that converts high voltage signals (±15 V) into TTL logic levels. Output voltages usually range from 0.2 V (low) to 4.0 V (high). The internal circuit functions as an on/off transistor. When the transistor base has a negative input voltage, the transistor is turned off and the collector voltage (the output of the MC 1489) is high. When the transistor base has a positive input voltage, the transistor is driven into saturation to 0.2 V.

**RS-232 Drivers/Receivers: MAX 232** One of the drawbacks of the line driver MC 1488 is that it requires additional voltages ±12 V from a power supply. Typically, such voltages are unavailable in systems compatible with the TTL logic. This difficulty can be resolved by using a specialized integrated device, MAX 232 (Figure 16.16(b)). It includes drivers and receivers (two each), and it generates ±10 V internally by using a voltage doubler and inverter circuits. The MAX 232 can replace the 1488 and 1489 and eliminate the need for ±12 V power supplies.

### 3. INITIALIZATION

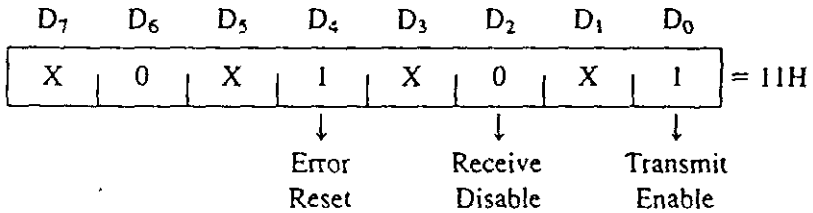
The control words necessary for the given specifications are as follows:

**Mode Word** Refer to Figure 16.15(a)



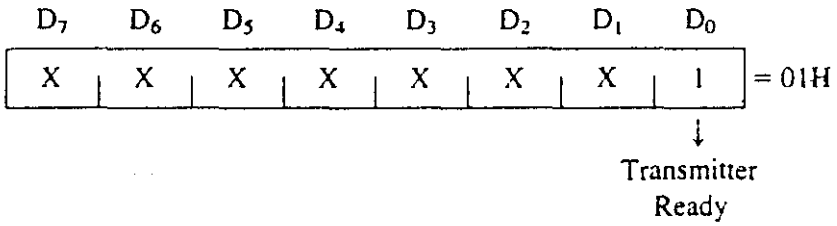

In a mode word, bits $D_1$ and $D_0$ can specify a baud factor that divides the clock frequency (TxC) to provide three different transmission rates. In this illustration, TxC is sixteen times the specified baud.

**Command Word** Refer to Figure 16.15(b)




In this command word, bit $D_0$ enables the transmitter, bit $D_4$ ignores any errors, and bit $D_6$ prevents reset of the 8251A; all other bits are don't care. In this illustration, bit $D_4$ also can assume don't care logic level.

**Status Word** Refer to Figure 16.15(c)




The MPU should check bit $D_0$ before transferring a character to the 8251A; it indicates the status of the pin TxRDY. When a byte is transferred from the transmitter buffer to the output register (see Figure 16.14 for the block diagram), bit $D_0$ is set to 1, and it is reset when the MPU loads the next byte in the buffer. Bit $D_2$ (TxEMPTY) indicates the status of the output register; this bit usually is not used except in applications such as half duplex mode.

## 4. INITIALIZATION INSTRUCTIONS

```
SETUP:    MOV   AL, 0CAH   ;Load the mode word
          OUT   0FEH, AL   ;Write mode word in control register
          MOV   AL, 11H    ;Load the command word to enable transmitter
          OUT   0FEH, AL   ;Enable the transmitter
STATUS:   IN    AL, 0FEH   :Read status register
          ROR   AL, 1      ;If D0 = 0, the transmitter buffer is full: go back
          JNC   STATUS     ;  and wait
```